ANSWERS

A12.6 DHCP, NAT and port forwarding

Networks and the web · A level · AQA 7517 4.9.4.7 · about 30 min

What this lesson is about

How a device gets its address, and how a router shares one public address with a whole network.

- DHCP
- NAT
- Port forwarding

Questions

6 marks in all

1. Put the DHCP messages in the order they are sent. [1 mark]

Number the lines 1 to 4 to put them in the right order.

___ `Offer`
___ `Acknowledge`
___ `Discover`
___ `Request`

Answer:

```
Discover
Offer
Request
Acknowledge
```

The client discovers a server, the server offers an address, the client requests it, and the server acknowledges the lease.

2. Which settings does a DHCP server typically give a device? Choose all that apply. [1 mark]

Tick every answer that is true.

- ☐ **A** IP address
- ☐ **B** Subnet mask
- ☐ **C** Default gateway
- ☐ **D** DNS server address
- ☐ **E** MAC address

Answer: A, B, C, D. The MAC address belongs to the device's network interface and is not assigned by DHCP.

3. When a packet leaves a home network through a router using NAT, what does the router change? [1 mark]

- ☐ **A** The private source IP address (and usually the source port) to its public address and a port of its own, recording it in a table
- ☐ **B** The destination IP address to a private address
- ☐ **C** The MAC address of the destination server
- ☐ **D** Nothing: NAT only affects incoming packets

> **Answer:** A. The router records the translation so it can change the destination of the reply back to the private socket.

4. Why is NAT used? Choose all that apply. [1 mark]

Tick every answer that is true.

- ☐ **A** Many devices can share one public IPv4 address
- ☐ **B** Private addresses are hidden from the Internet
- ☐ **C** It speeds up DNS lookups
- ☐ **D** It gives every device a globally unique address

> **Answer:** A, B. NAT conserves public IPv4 addresses and hides the internal network; unique addresses for every device is what IPv6 offers instead.

5. A student runs a game server at 192.168.1.40 on their home network. What lets friends on the Internet connect to it? [1 mark]

- ☐ **A** A port forwarding rule mapping a public port on the router to 192.168.1.40 and the server's port
- ☐ **B** Giving the server a DHCP lease
- ☐ **C** Turning off the router's DNS
- ☐ **D** Using a thin client

> **Answer:** A. Without port forwarding, a connection started from outside has no NAT table entry and is dropped.

6. What does this program print? [1 mark]

```
table = {}
next_port = 50000
for socket in ['192.168.1.5:3000', '192.168.1.6:3000', '192.168.1.5:3000']:
    if socket not in table:
        table[socket] = next_port
        next_port = next_port + 1
    print(socket, '->', table[socket])
```

> **Answer:**
>
> ```
> 192.168.1.5:3000 -> 50000
> 192.168.1.6:3000 -> 50001
> 192.168.1.5:3000 -> 50000
> ```
>
> Each new private socket gets the next public port; the repeated socket reuses its existing entry.

The task: get an address

The Router on this mat is a DHCP server that talks over the radio in a simplified form of DORA. Messages are words separated by single spaces. 1. Send `DISCOVER`. 2. The Router replies `OFFER <address> <mask> <gateway> <lease seconds>`. Print `offered <address>`. 3. Send `REQUEST <address>`, using the address from the offer. 4. The Router replies `ACK <address> <mask> <gateway> <lease seconds>`. From the ACK, print `address: <address>`, then `gateway: <gateway>`, then `lease: <hours> hours`, where hours is the lease in seconds divided by 3600 as a whole number. Check `messages()` every 0.1 s while you wait for each reply. Read every value from the replies; the robot does not drive.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

send("DISCOVER")
wait(1)
print(messages())
```

The hint students can ask for: Write one helper that waits for a reply starting with a given word and hands back its fields. Use it after each message you send. The offer tells you what to request; only the acknowledgement makes the address yours, so print the settings from that.

A solution

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def wait_for(word):
    while True:
        wait(0.1)
        for sender, text in messages():
            fields = text.split()
            if fields[0] == word:
                return fields

send("DISCOVER")
offer = wait_for("OFFER")
print("offered", offer[1])
send("REQUEST " + offer[1])
ack = wait_for("ACK")
print("address:", ack[1])
print("gateway:", ack[3])
print("lease:", int(ack[4]) // 3600, "hours")
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.